Compact系列CPLD器件从SPI接口配置和编程应用指南

(AN03009，V1.0)

(2021.8.24)

深圳市紫光同创电子有限公司

文档版本修订记录

版本号	发布日期	修订记录
V0.1	2020-2-24	初始版本
V1.0	2021-8-24	1、文档介绍中删除具体器件描述； 2、合并原 2、3 章； 3、修改描述特征控制位描述，删除无关位； 4、修改图 1，体现 CCS； 5、修改表 21，删除 Package、Bank 列； 6、修改设置特征控制位描述位置，以及体现新软件设置界面； 7、修改表 2，增加指令操作对象列； 8、增加图 2、图 3 注释； 9、增加关断重配章节； 10、增加编程嵌入式 Flash 章节；

名词术语解释

Abbreviations 缩略语	Full Spelling 英文全拼	Chinese Explanation 中文解释
JTAG	Joint Test Action Group	联合测试工作组
CCS	Configuration Control System	配置控制系统
SPI	Serial Peripheral Interface	串行外围接口
UID	Unique ID	唯一的 ID

目录

1 文档介绍 7
2 配置细节 8
2.1 从SPI接口介绍 8
2.2 特征控制位 10
2.3 从SPI指令集 11
2.4 配置流程 12
2.4.1 配置/重配 12
2.4.2 关断重配 16
2.5 编程嵌入式Flash 19
2.5.1 PROGRAM时序 22
2.5.2 Address 22
3 配置案例 24
3.1 硬件连接 24
3.2 设置特征控制位 25
3.3 工程说明 26
3.4 案例演示 27
3.5 函数说明 28
3.5.1 底层函数 28
3.5.1.1 RDID 28
3.5.1.2 RDUID 28
3.5.1.3 RDSR 29
3.5.1.4 PROGRAM 30
3.5.2 应用函数 30
3.5.2.1 唤醒嵌入式FLASH 31
3.5.2.2 休眠嵌入式FLASH 31
3.5.2.3 读UID 31
3.5.2.4 编程主自加载数据流 32
3.5.2.5 擦除主自加载数据流 32
4 附录 33
免责声明 36

图目录

图 1 从SPI应用示例 8
图 2 配置/重配流程图 13
图 3 配置/重配流程时序 14
图 4 关断重配流程 17
图 5 关断重配流程时序 18
图 6 编程嵌入式FLASH 1 19
图 7 编程嵌入式FLASH 2 20
图 8 关断重配流程时序 21
图 9 1K/2K/4K/7K PROGRAM时序 22
图 10 10K PROGRAM时序 22
图 11 案例硬件电路图 24
图 12 位流设置特征控制位 25
图 13 直接修改特征控制位 26
图 14 串口调试情况 27
图 15 RDID时序图 28
图 16 RDUID时序图 29
图 17 RDSR时序图 30
图 18 PROGRAM时序图 30

表目录

表 1 从SPI引脚定义 8
表 2 配置CPLD相关的从SPI指令集 11
表 3 从SPI的关断重配命令流程 16
表 4 嵌入式FLASH普通存储器存储空间 22
表 5 1K/2K/4K/7K ADDRESS含义 23
表 6 10K ADDRESS含义 23
表 7 SPI引脚连接关系 24
表 8 FLASH芯片SPI引脚 25
表 9 工程概述 26
表 10 工程目录分类 26
表 11 串口配置参数 27
表 12 读DVICEID函数 28
表 13 读UID 28
表 14 读状态寄存器 29
表 15 向嵌入式FLASH写入数据 30
表 16 唤醒嵌入式FLASH 31
表 17 休眠嵌入式FLASH 31
表 18 读UID 31
表 19 编程主自加载位流 32
表 20 擦除主自加载位流 32
表 21 COMPACT系列CPLD的SPI接口编号 33

1 文档介绍

本文档主要介绍紫光同创CPLD从SPI配置模式应用特点、从SPI配置所需要的软硬件环境、CPLD内部特殊寄存器设置等。文档最后以MCU配置CPLD的实际应用方案为例，为用户提供一个完整配置流程参考。

2 配置细节

2.1 从SPI接口介绍

通过从SPI接口，用户可以配置/重配CRAM、编程嵌入式FLASH、读写特征控制位等。

从SPI接口为用户配置提供一种方便而通用的方式。通常可由CPU、MCU、DSP等器件作为主设备，CPLD作为从设备，这种应用方式可由下图表示：

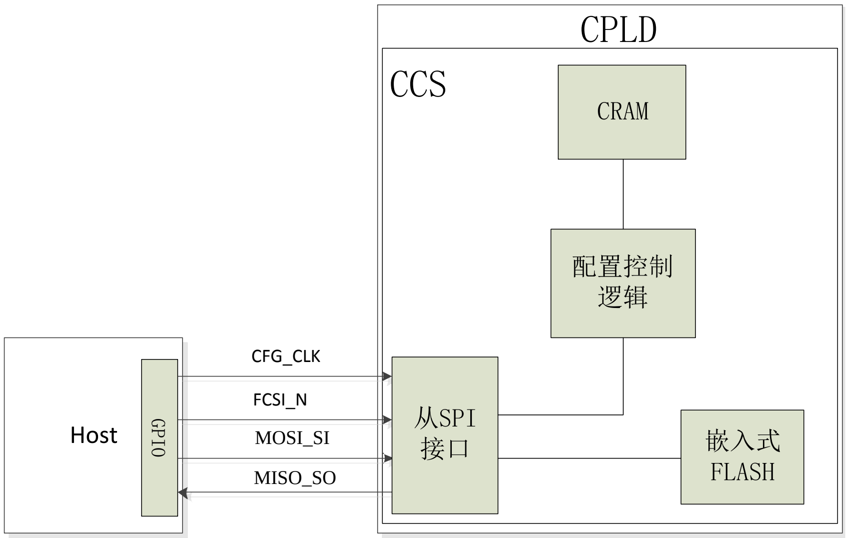


图 1 从SPI应用示例

其中从SPI接口引脚定义如下表所示：

表 1 从SPI引脚定义

引脚名称	I/O	描述
CFG_CLK	I	配置时钟，写频率最高100MHz，读频率最高10MHz，
FCSI_N	I	片选信号，低有效，CFG_CLK上升沿采样
MOSI_SI	I	串行数据输入信号，CFG_CLK上升沿采样
MISO_SO	O	串行数据输出信号，CFG_CLK下降沿发送

当使用从SPI接口时，时钟CFG_CLK由主设备提供。当使用从SPI接口读出数据时（如读IDCODE、特征控制位、位流等），CFG_CLK频率不能超过10MHz，当通过从SPI接口写入数据时，频率不能超过100MHz。

从SPI接口片选信号FCSI_N为同步信号，在CFG_CLK上升沿采样，当FCSI_N为高时，MISO_SO为高阻态。当FCSI_N为低时，CPLD可通过MOSI_SI采样数据，或者通过MISO_SO输出数据。Compact系列CPLD不同型号、不同封装的器件的从SPI接口在引脚上的分布不同，具体可通过表 21或对应器件的封装手册获取。

Compact系列CPLD的从SPI接口是否作为配置接口，以及配置完成后，是否保留作为配置接口，是通过芯片的特征控制位进行控制，见特征控制位。

2.2 特征控制位

Compact系列CPLD的特征控制位存储在非易失存储单元嵌入式FLASH中，在每次芯片上电时，会由芯片CCS读出。CCS根据特征控制位选择相应复用配置管脚的功能。

CPLD包含由32位特征控制位组成的特征控制组CTL。特征控制位详细信息参考文档《UG030004_Compact系列CPLD配置（Configuration）用户指南》。

从SPI接口的使能由特征控制位中的cfg_sspi_en_n（CTL[6]）和persist_sspi_n（CTL[15]）控制。当cfg_sspi_en_n（CTL[6]）为0时，使能配置模式下从SPI接口，当persist_sspi_n（CTL[15]）为0时，使能用户模式下从SPI接口。值得注意的是，从SPI接口与主SPI接口共用CFG_CLK、MISO_SO、MOSI_SI。主SPI与从SPI虽然可以同时使能（主SPI由cfg_mspi_en（CTL[4]）和persist_mspi（CTL[28]）控制），但是在同一时刻，只能有一种接口在使用。建议不同时使能主SPI和从SPI。若同时使能时，应当注意以下几种时刻，主SPI接口处于工作状态，用户不可操作从SPI接口：

1. 芯片上电后，主动通过主 SPI 接口从外部 SPI FLASH 中获取位流，直到芯片配置完成或读完 SPI FLASH。
2. 用户模式下同时使能保留主 SPI 和从 SPI 接口，并触发自动软纠错。

芯片特征控制位可通过以下几种方式进行修改：

1. PDS（Pango Design Suit）下载工具（Fabric Configuration）下载sbit文件时，会将sbit文件中包含的特征控制位写到芯片中。
2. PDS下载工具中单独编程器件特征控制位。
3. 在使能JTAG接口前提下，使用JTAG接口PROGRAM_CTL指令。
4. 在使能从SPI接口前提下，使用从SPI接口PROGRAM_CTL指令。
5. 在使能从IIC接口前提下，使用从IIC接口PROGRAM_CTL指令。
6. 使用内部从APB接口PROGRAM_CTL指令。

2.3 从SPI指令集

表 2 配置CPLD相关的从SPI指令集

指令	描述	操作码	操作对象
NOP	No Operation	FF	
RDID	Read Identification	A1	
RDUSER	Read Usercode	A2	
RDSR	Read Status Register	A3	
RDUID	Read Unique Identification	A4	
RDLOCK	Read Embedded Flash Lock Information	A5	
CFG	Config Bitstream	50	CRAM
WREN	Write Enable	51	
WRDIS	Write Disable	52	
RESET	Reset CPLD	60	
ERASE	Erase Bulk	10	EFlash
ERASE_PAGE	Erase Page	11	EFlash
ERASE_CTL	Erase Feature Control Page	12	EFlash（Feature Control）
PROGRAM	Program Page	20	EFlash
PROGRAM_CTL	Program Feature Control Page	22	EFlash（Feature Control）
READ	Read	30	EFlash
READ_CTL	Read Feature Control Page	31	EFlash（Feature Control）
PROGRAM_LOCK	Lock Embedded Flash	40	EFlash
EFlash_SLEEP	Embedded Flash Sleep	70	EFlash
EFlash_WAKEUP	Embedded Flash Wake Up	71	EFlash

指令与指令之间需要拉高FCSI_N至少一个周期，指令均为高位先进。

2.4 配置流程

2.4.1 配置/重配

配置，CPLD 芯片从开始配置到进入用户模式的过程；重配，CPLD芯片进入用户模式后，重新复位（指令复位或外部触发RSTN硬复位），清空所有CRAM，退出用户模式，然后进行重配置的过程。流程如下：

1. 上电
2. 待INIT_FLAG_N的值为1后，写RDID指令，读出器件IDCODE
3. 检查IDCODE的值
4. 如果IDCODE不匹配，终止操作。如果IDCODE匹配，写WREN指令
5. 写入CFG指令，载入位流
6. 位流发送完成后，如果用户模式下从SPI/从I^2C接口释放给用户使用（特征控制位persist_sspi_n/persist_si2c_n为1），察看INIT_FLAG_N和CFG_DONE的值，操作结束。如果用户模式下从SPI接口做为配置接口使用，写入WRDIS指令
7. 写入RDSR指令，读出器件状态寄存器

上电后的任何时刻，可以对CPLD器件进行冷启动（将RSTN置0）。冷启动后，可以进入步骤2，对CPLD器件进行重配。

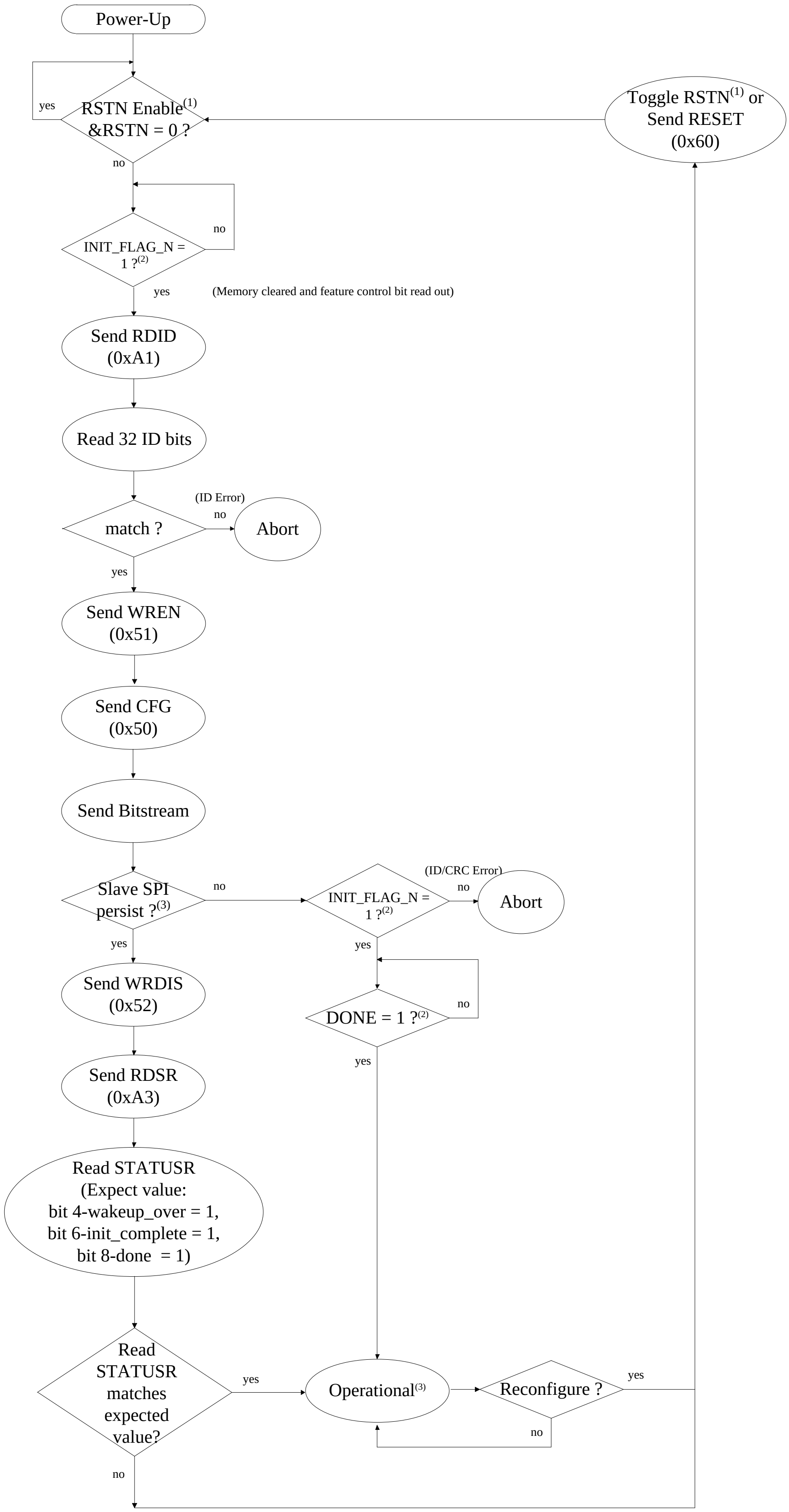

Power-Up
yes
RSTN Enable(1)
&RSTN = 0 ?
no
Toggle RSTN(1) or
Send RESET
(0x60)
no
INIT_FLAG_N =
1 ?(2)
yes
(Memory cleared and feature control bit read out)
Send RDID
(0xA1)
Read 32 ID bits
(ID Error)
no
match ?
Abort
yes
Send WREN
(0x51)
Send CFG
(0x50)
Send Bitstream
Slave SPI
persist ?(3)
no
(ID/CRC Error)
INIT_FLAG_N =
1 ?(2)
no
Abort
yes
Send WRDIS
(0x52)
yes
DONE = 1 ?(2)
no
yes
Send RDSR
(0xA3)
Read STATUSR
(Expect value:
bit 4-wakeup_over = 1,
bit 6-init_complete = 1,
bit 8-done = 1)
Read
STATUSR
matches
expected
value?
yes
Operational(3)
Reconfigure ?
yes
no
no

图 2 配置/重配流程图

注：

（1） 若使能RSTN，则需要RSTN拉高后，器件才开始初始化。

（2） 若使能INIT_FLAG_N，则该引脚可用于指示初始化完成以及加载是否有误；而DONE使能后可用于指示加载完成。若同时使能RSTN，则对这两个引脚的判断需要在RSTN释放（拉高）之后，在RSTN为低时这两个引脚可能无法正确指示。

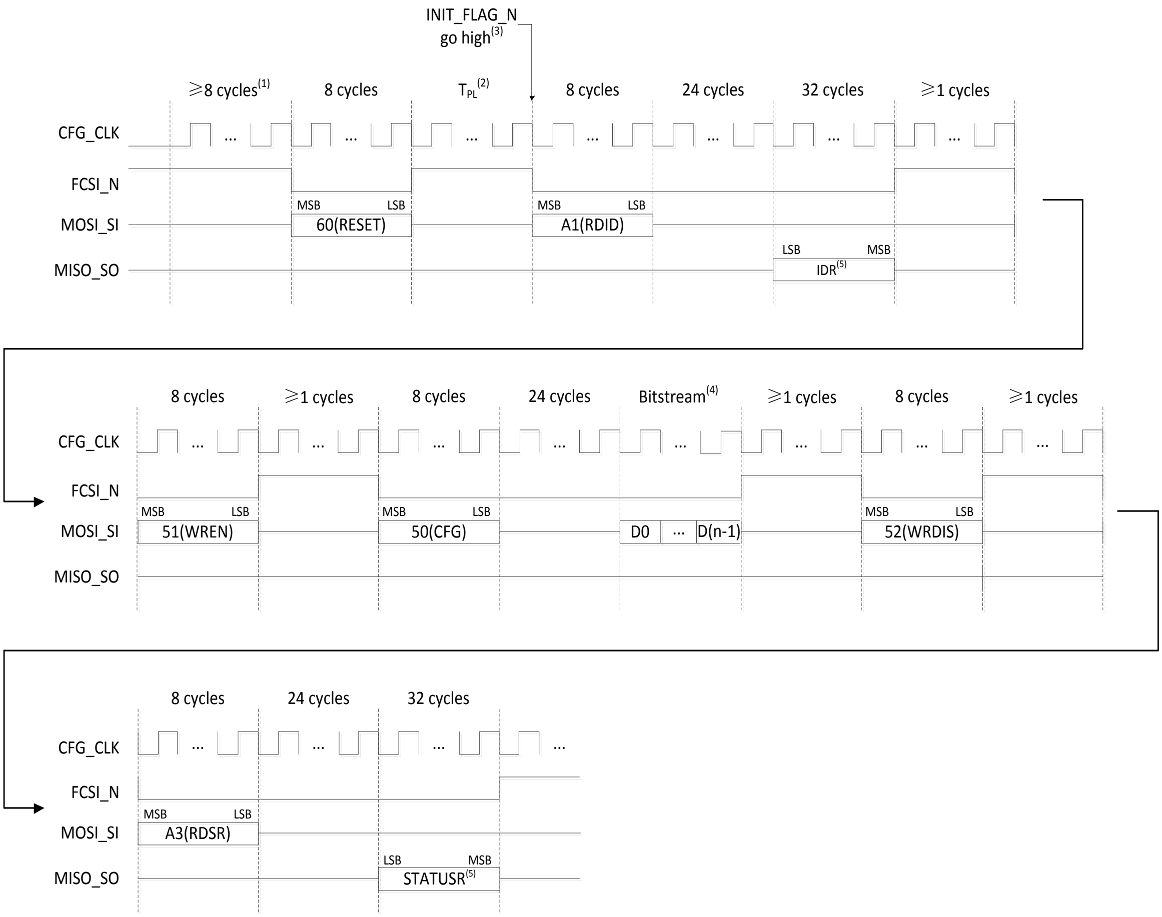


图 3 配置/重配流程时序

注：

（1） 上电后第一个指令前需要在FCSI_N为高时给至少8个时钟，否则第一条指令无法工作。

（2） T_{PL}见数据手册。

（3） 若使能INIT_FLAG_N，则该引脚可指示初始化完成。该例子为指令复位，若使用硬复位RSTN的方式，则需确保RSTN释放后，才可判断INIT_FLAG_N。

（4） 此处可使用位流格式有bin（建议）、sbit、sfc，这几种文件均为二进制流，发送时只需从头发到尾即可。如位流的第一个字节为0xAA（1010_1010），则D0=1，D1=0，……。

（5） IDR即器件标识寄存器，该寄存器值即Device ID；STATUSR即状态寄存器。寄存器详情见配置UG。读出寄存器值时，是低位先出，如ID为0x0049_9899，则移出顺序为1、0、0、1、1……。

（6） 除非特殊说明，任何一条指令流程中（包括dummy及数据），FCSI_N不能中途拉高，拉高就意味着该行为终止。

2.4.2 关断重配

关断重配，不重新复位，不清空所有CRAM，未退出用户模式，关断器件进行重配置的过程。关断重配需要通过CFG指令向CCS写一连串命令来实现，这些命令按照位流包格式，完成对配置寄存器的修改，以实现关断重配功能。

表 3 从SPI的关断重配命令流程

流程	数据
写CFG指令，载入：	0x50
100个填充字	0xFFFFFFFF …… 0xFFFFFFFF
同步字	0x01332D94
空操作	0xA0000000
写RSTCRC命令到CMDR寄存器	0xA8800001 0x00000001
写OPTION0R寄存器（选择唤醒关断时钟）	0xAE400001 0x00000100
写CMASKR寄存器	0xADC00001
写CTRLR寄存器（禁止屏蔽可变存储单元回读）	0x00000010
写SWAKEDOWN命令到CMDR寄存器	0xA8800001 0x00000008
20个空操作	0xA0000000 …… 0xA0000000
写GDOWN命令到CMDR寄存器	0xA8800001 0x0000000A
写DESYNC命令到CMDR寄存器	0xA8800001 0x0000000B
位流	位流

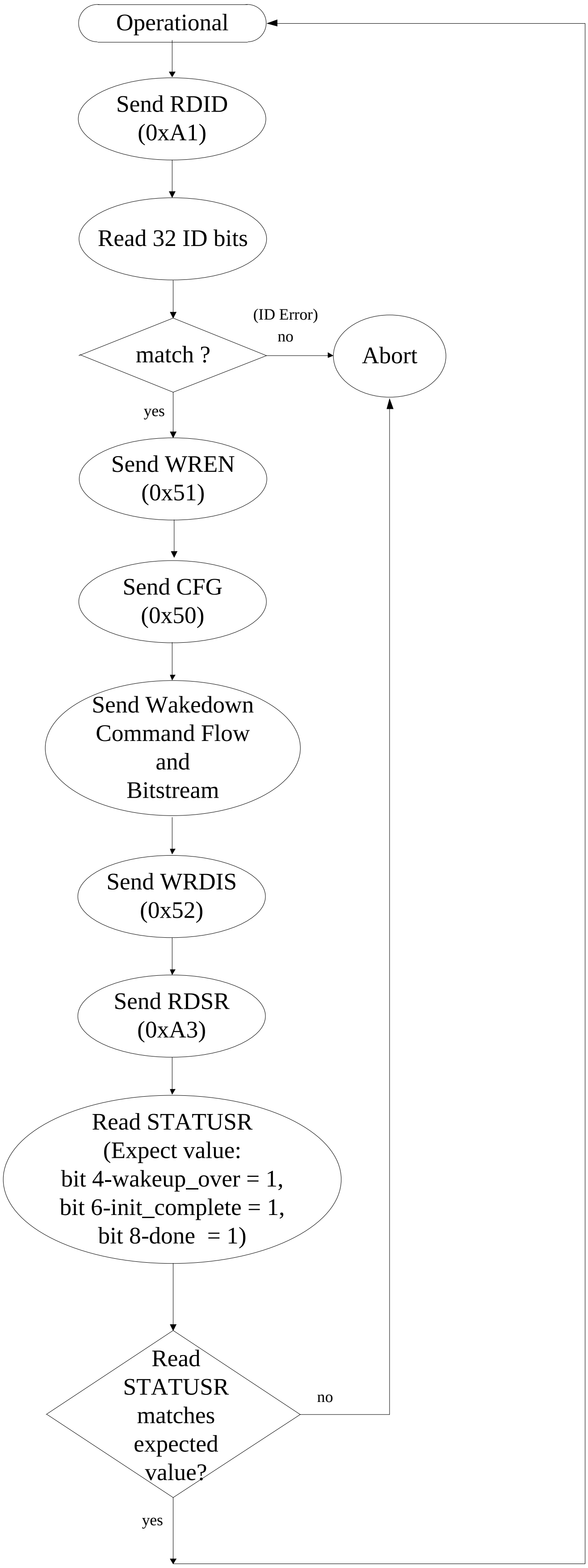

Operational
Send RDID
(0xA1)
Read 32 ID bits
match ?
(ID Error)
no
Abort
yes
Send WREN
(0x51)
Send CFG
(0x50)
Send Wakedown
Command Flow
and
Bitstream
Send WRDIS
(0x52)
Send RDSR
(0xA3)
Read STATUSR
(Expect value:
bit 4-wakeup_over = 1,
bit 6-init_complete = 1,
bit 8-done = 1)
Read
STATUSR
matches
expected
value?
no
yes

图 4 关断重配流程

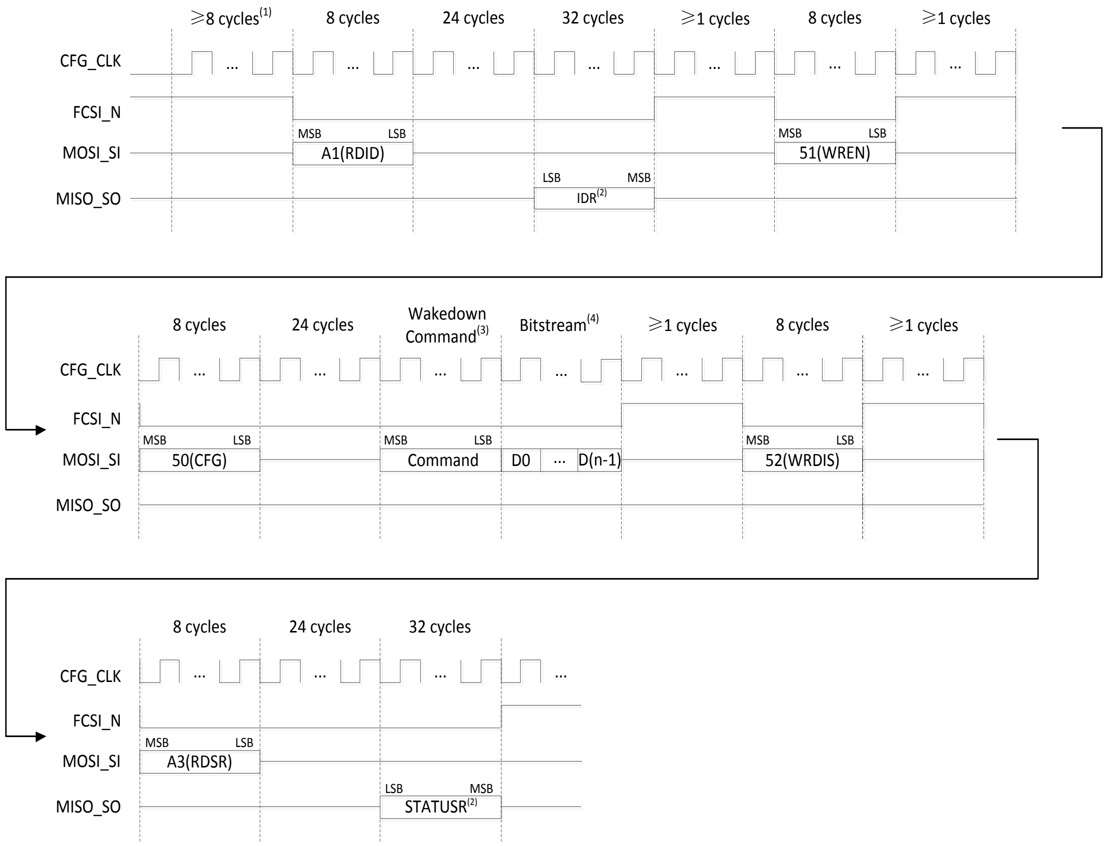


图 5 关断重配流程时序

注：

（1） 上电后第一个指令前需要在FCSI_N为高时给至少8个时钟，否则第一条指令无法工作。

（2） IDR即器件标识寄存器，该寄存器值即Device ID；STATUSR即状态寄存器。寄存器详情见配置UG。读出寄存器值时，是低位先出，如ID为0x0049_9899，则移出顺序为1、0、0、1、1……。

（3） 此处关断指令流为高位先进，如其中的0x01332D94（即二进制0000_0001_0011…），则移入数据顺序为0、0、0、0、0、0、0、1、0……。

（4） 此处可使用位流格式有bin（建议）、sbit、sfc，这几种文件均为二进制流，发送时只需从头发到尾即可。如位流的第一个字节为0xAA（1010_1010），则D0=1，D1=0，……。

（5） 除非特殊说明，任何一条指令流程中（包括dummy及数据），FCSI_N不能中途拉高，拉高就意味着该行为终止。

2.5 编程嵌入式Flash

嵌入式Flash支持直接通过JTAG，从SPI，从I^{2}C接口直接对嵌入式Flash进行操作。本节介绍从SPI的操作流程。

从SPI编程嵌入式Flash的完整流程如下所示：

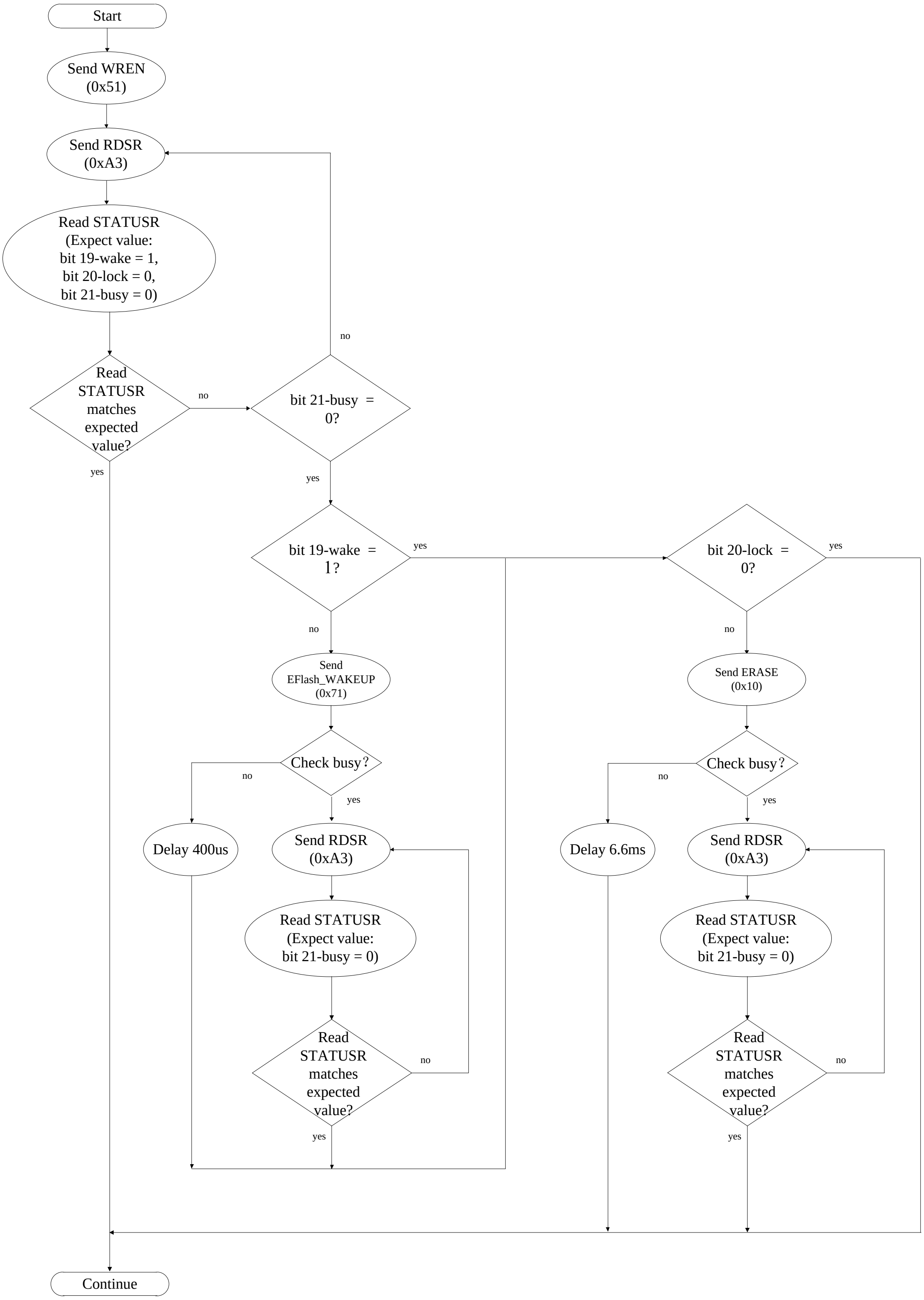


图 6 编程嵌入式Flash 1

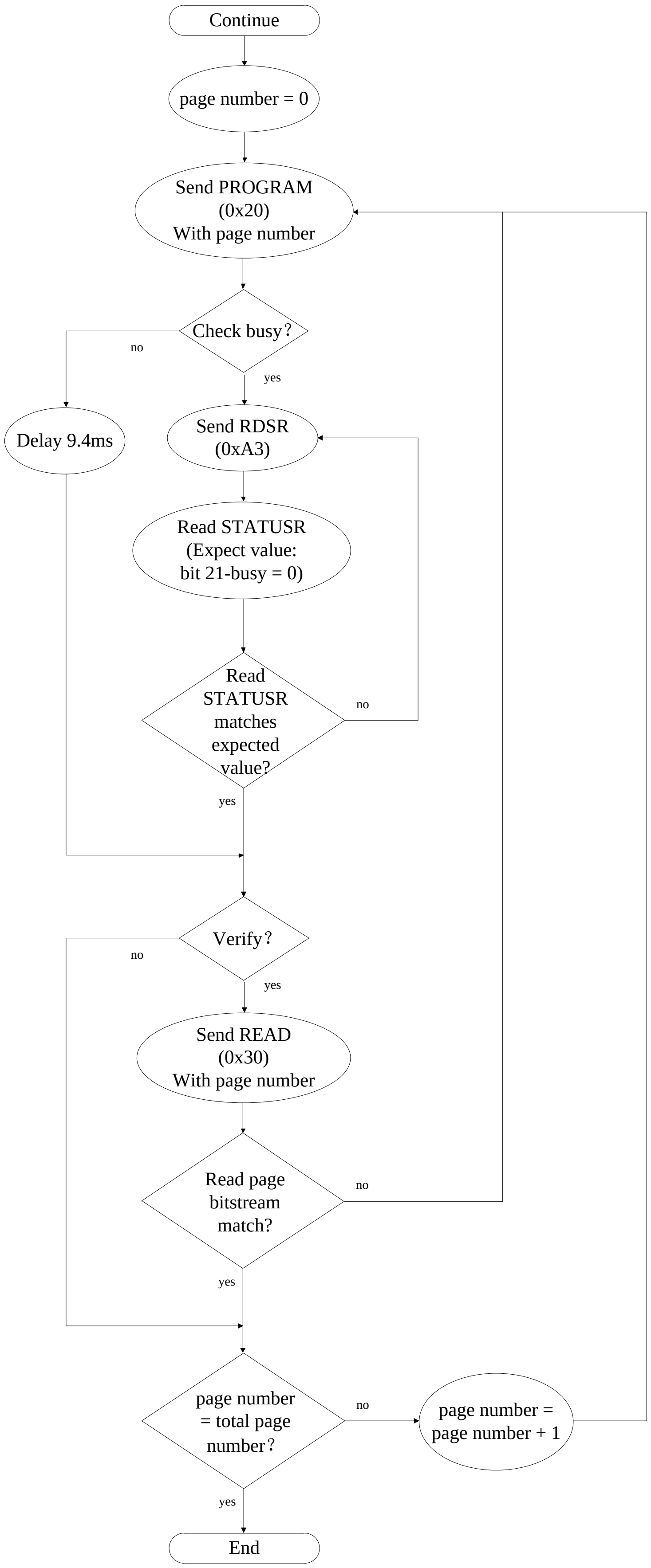


图 7 编程嵌入式Flash 2

编程时序举例如下图所示，其中假设器件已唤醒并且EFlash不处于Lock状态。

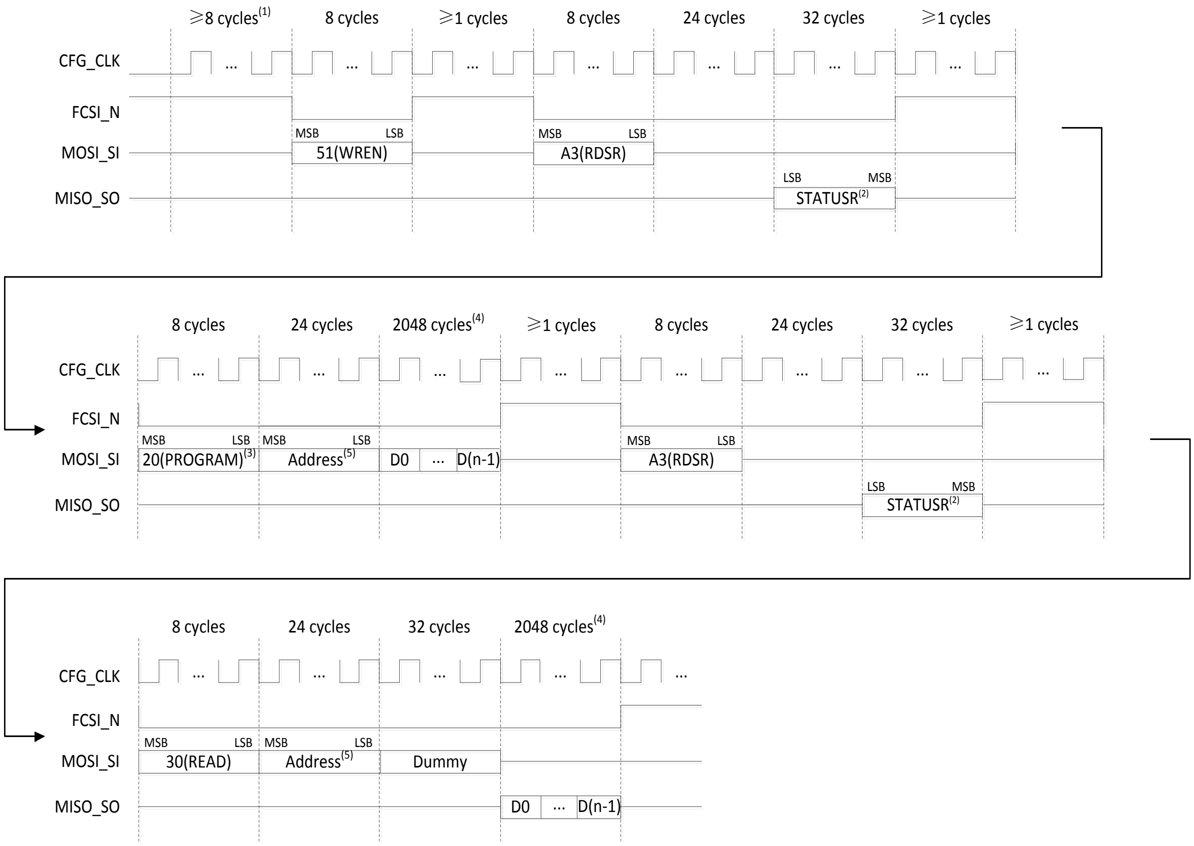


图 8 关断重配流程时序

注：

（1） 上电后第一个指令前需要在FCSI_N为高时给至少8个时钟，否则第一条指令无法工作。

（2） STATUSR即状态寄存器。寄存器详情见配置UG。读出寄存器值时，是低位先出。

（3） PROGRAM指令时序对不同器件有所不同，详见PROGRAM时序。

（4） 此处按页操作，每页有256Bytes，数据为高位先进/先出。

（5） Address对应关系对不同器件有所不同，详见Address。

（6） 除非特殊说明，任何一条指令流程中（包括dummy及数据），FCSI_N不能中途拉高，拉高就意味着该行为终止。

2.5.1 PROGRAM时序

PROGRAM 为编程指令。

用于编程嵌入式 Flash 普通页，不同器件时序有所不同。

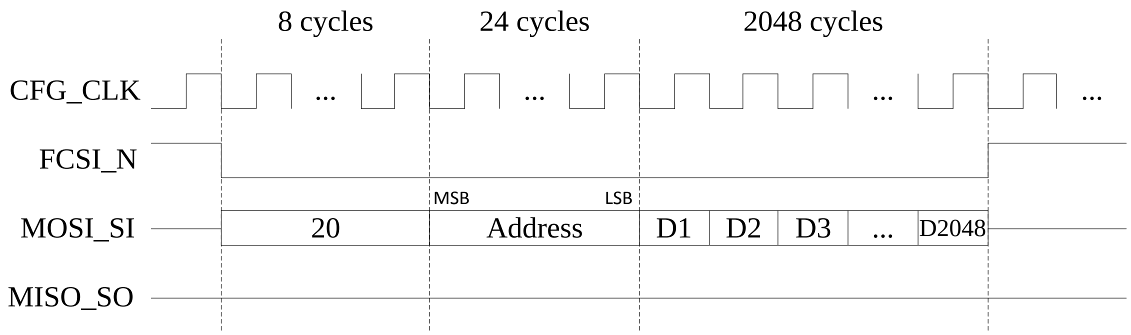


图 9 1K/2K/4K/7K PROGRAM时序

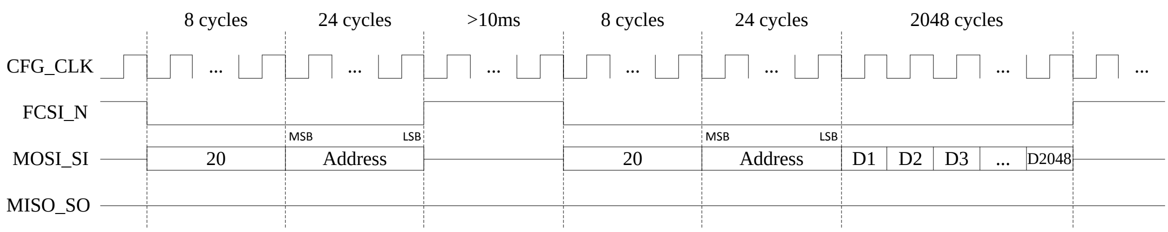


图 10 10K PROGRAM时序

2.5.2 Address

PGC的EFlash最小寻址单位为32bit字，每页包含256Bytes，即64个字。不同器件包含的存储容量不同，因此总页数也不同。对于较大的器件，EFlash分为多个堆，因此Address不连续。具体器件存储空间及地址映射如下所示。

表 4 嵌入式Flash普通存储器存储空间

器件	堆数	每个堆的页数	页大小（Byte）	总容量（Byte）	嵌入式Flash总页数
PGC1KG/L	1	332	256	84992	332
PGC2KG/L	1	332	256	84992	332
PGC4KD/L	4	320	256	327680	1280
PGC7KD	4	452	256	462848	1808
PGC10KD	4	640	256	655360	2560

表 5 1K/2K/4K/7K Address含义

Address	名称	描述
[23:19]	保留	需要设置为 5’b00000
[18:17]	Ba[1:0]	堆地址 对于 1K、2K 器件为保留位，其中 2K 器件需设置为 2’b11
[16:8]	Ra[8:0]	页地址
[7:6]	保留	需要设置为 2’b00
[5:0]	Ca[5:0]	32 位数据页内偏移地址 对于 PROGRAM 指令，需设置为 6’b000000

表 6 10K Address含义

Address	名称	描述
[23:20]	保留	需要设置为 5’b00000
[19:18]	Ba[1:0]	堆地址
[17:8]	Ra[8:0]	页地址
[7:6]	保留	需要设置为 2’b00
[5:0]	Ca[5:0]	32 位数据页内偏移地址 对于 PROGRAM 指令，需设置为 6’b000000

3 配置案例

本节内容以STM32F103控制读写PGC的嵌入式FLASH以及加载CRAM为例介绍PGC从SPI接口配置的应用。

本案例基于野火STM32F103MINI的硬件平台，软件开发环境为Keil5。

3.1 硬件连接

硬件连接包含两个部分，STM32与CPLD之间的SPI连接、STM32与FLASH芯片的硬件连接，前者需要使用杜邦线连接，后者为PCB电路连接（野火STM32MINI 板载STM32F103和W25Q64芯片）。

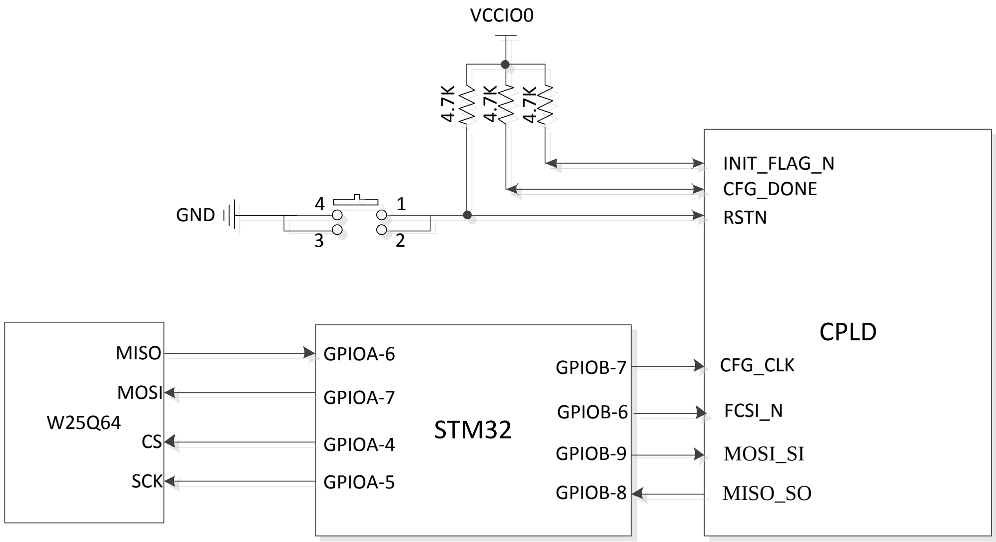


图 11 案例硬件电路图

配置过程中，需要保持模拟SPI接口的单片机GPIO Pin与PGC2K的SPI接口一直处于连接状态。

表 7 SPI引脚连接关系

STM32引脚	PGC2KG引脚
GPIOB-6	FCSI_N
GPIOB-7	CFG_CLK
GPIOB-8	MISO_SO
GPIOB-9	MOSI_SI

案例程序中，PGC2K的位流存放于STM32电路板的FLASH芯片上，FLASH芯片使用标准SPI接口，单片机与FLASH芯片的引脚连接关系如下表所示.

表 8 FLASH芯片SPI引脚

FLASH芯片引脚	STM32引脚
CS	GPIOA-4
SCK	GPIOA-5
MISO	GPIOA-6
MOSI	GPIOA-7

下载演示位流的时候，需要把表5中的FLASH引脚接到外部，从地址零开始写入PGC2K的位流，需要注意的是，烧录FLASH过程中，这个板子需要一直按下复位键。下载FLASH可以使用Pango Fabric Configuration，只需要把这些对应的FLASH引脚与Pango USB Cable连接。

3.2 设置特征控制位

如章节2.2所述，特征控制位控制从SPI是否使能，若要使用从SPI，需确保特征控制位设置使能从SPI接口。芯片出厂时默认为使能从SPI，若不慎关闭，可通过预留的其他接口如JTAG、从I2C接口修改特征控制位进行打开。以下介绍PDS下载工具使用JTAG接口打开从SPI的方法。

加载CRAM时，会默认烧写特征控制位。在位流设置中打开从SPI（选择Dedicated IO），并使用下载工具Fabric Configuration下载该位流，则从SPI接口打开。

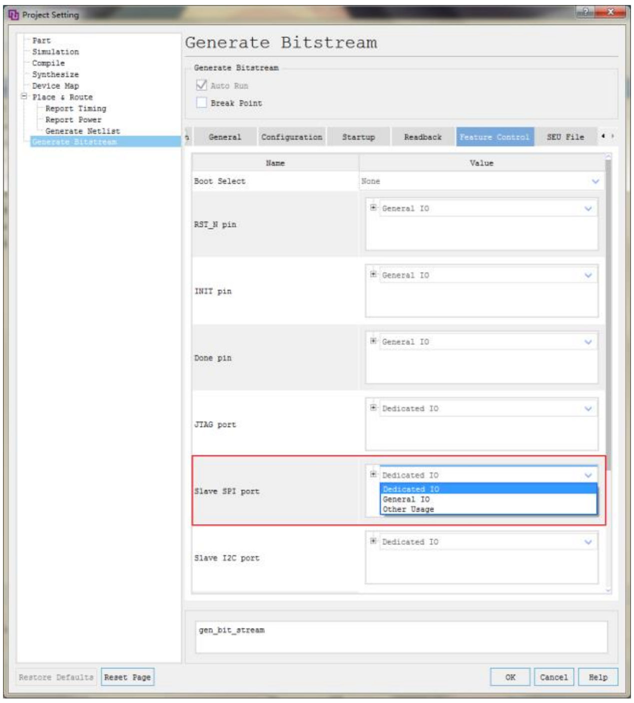


图 12 位流设置特征控制位

或者可使用下载工具Fabric Configuration直接修改特征控制位(取消勾选下图红框选项)，

并复位或重新上电生效。

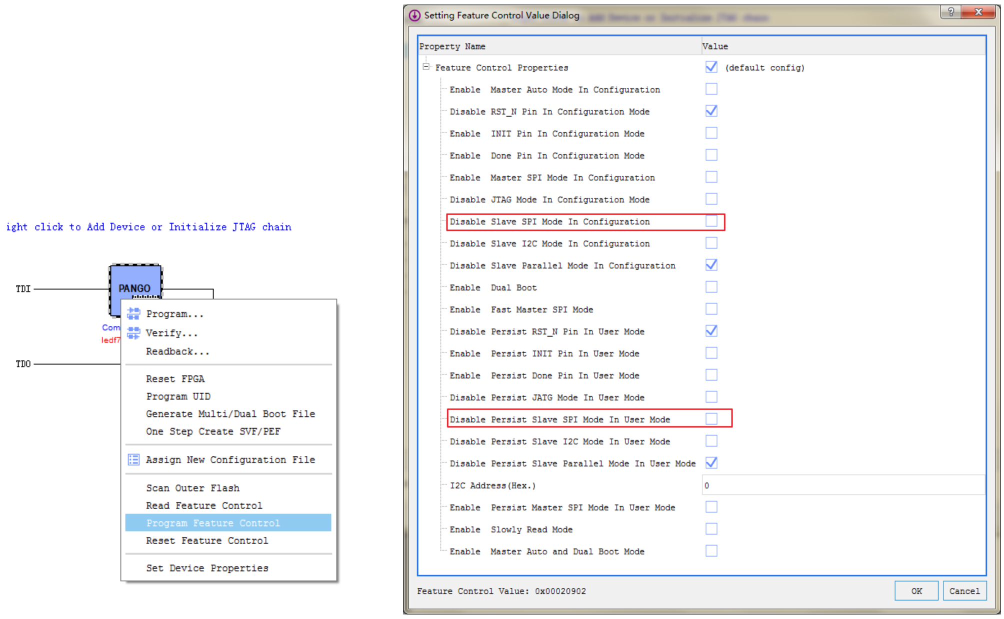


图 13 直接修改特征控制位

3.3 工程说明

表 9 工程概述

应用范围	
支持器件	例程由STM32F103运行，稍作更改即可移植到其他单片平台
使用语言	C
提供的设计文件	
示例电路板的电路图	PDF文件
C语言源程序	.c和.h文件
keil5工程文件	RAR压缩文件
开发者工具支持	
设计工具	Keil5 uVision5版本

工程目录的用途如下：

表 10 工程目录分类

DOC	记录文档
Libraries	STM32的标准库文件
Listing	Keil5输出的listing相关的文件

Output	Keil5编译产生中间文件等
Project	Keil5的project文件放置的位置
User	程序源代码存放路径

3.4 案例演示

Demo程序是一个基于串口控制并进行数据交互的一个例程，用户通过串口输入指令控制嵌入式系统，从而实现对CPLD的配置，使用时，需要安装串口调试助手，调试助手的配置如下表所示。

表 11 串口配置参数

配置项	参数
波特率	115200
校验位	None
数据位	8
停止位	1
十六进制显示	关闭
十六进制发送	关闭

复位STM32后，串口调试助手的显示如图所示，发送框输入对应功能的数字，并发送，单片机就会执行相应操作。



图 14 串口调试情况

3.5 函数说明

3.5.1 底层函数

这类函数比较精简，属于比较底层的函数，通常需要互相配合才能使用，并通过额外调用串口输出或者其他途径才能观察到运行结果，这类函数与《Compact 系列 CPLD配置（configuration）用户指南》中附录3功能相对应。

3.5.1.1 RDID

表 12 读DviceID函数

函数名	unsigned int SSPI_RDID(void);
功能描述	用于读取芯片的芯片ID号，返回值即为芯片的ID号，该ID号与芯片的规格型号对应 最新对应表格请咨询供应商
输入参数说明	
返回值	无
使用案例	参考SSPI_FPGA_Init()
实际应用场景	识别当前芯片型号是否与位流一致

用于回读CPLD器件器件的IDCODE，低位先出，时序图如下图所示。

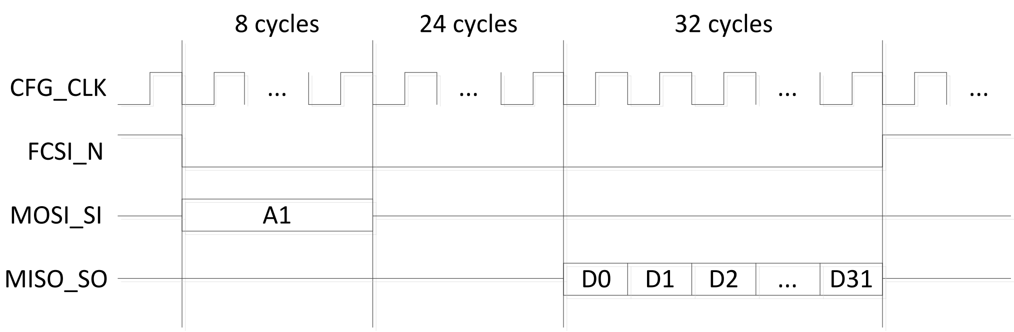


图 15 RDID时序图

3.5.1.2 RDUID

表 13 读UID

函数名	unsigned int PG_SSPI_ReadOutUIDLow(void); unsigned int PG_SSPI_ReadOutUIDHigh(void)
功能描述	用于读取芯片的UID，由于芯片的UID为64位，使用两个函数分别读取高32位和低32位
输入参数说明	
返回值	无

使用案例	参考PGC2KG_demo.h中 PGC2K_ReadAllUid(PGC2KDevice ***dev**)

用于回读CPLD器件器件的UID，低位先出，时序图如下图所示。

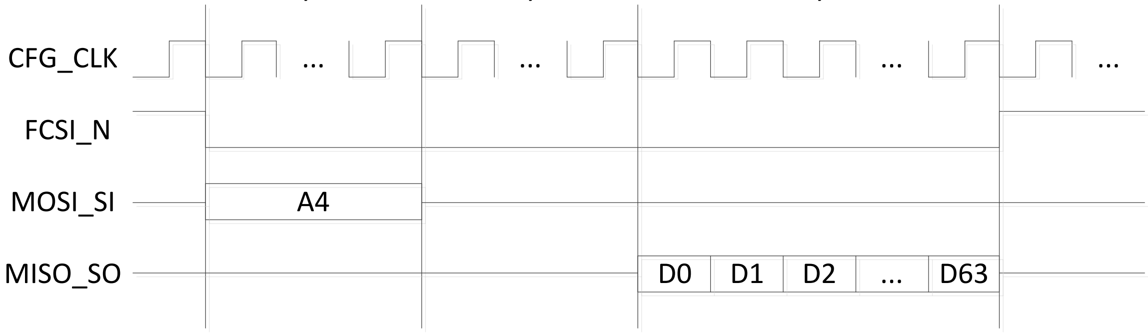


图 16 RDUID时序图

3.5.1.3 RDSR

表 14 读状态寄存器

函数名	unsigned int SSPI_RDSR(void);
功能描述	建议在唤醒嵌入式FLASH后使用，用于读取当前特征控制位
输入参数说明	无
返回值	当前芯片的特征控制位
使用案例	SSPI_IsEFLASHBusy(void)
说明	在使用RDSR，可以在输入完 ERASE PROGRAM READ PROGRAM_LOCK ERASE_PAGE 等指令后，直接使用，检测当前嵌入式FLASH的状态，不需要执行WRDIS后才使用

用于回读状态寄存器，低位先出，时序图如下图所示。

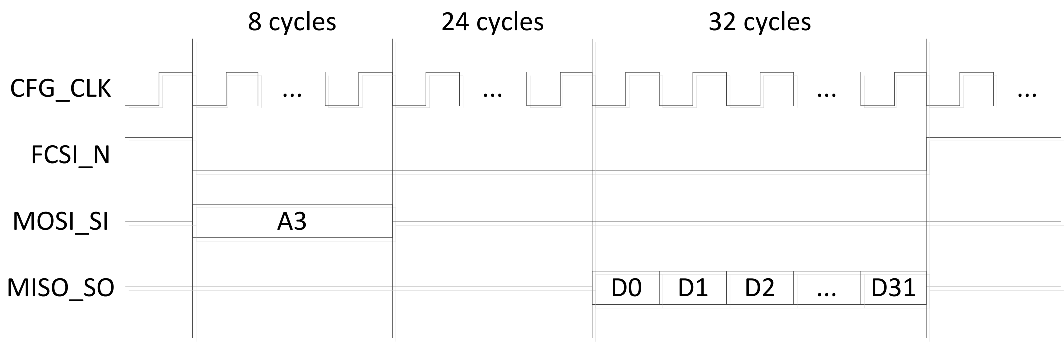


图 17 RDSR时序图

3.5.1.4 PROGRAM

表 15 向嵌入式FLASH写入数据

函数名	SSPI_Write(unsigned int addr,unsigned char *dataoutput)
功能描述	向指定地址的EFLASH中写入一页数据
输入参数说明	addr：写入页的地址 *dataoutput ：待写入数据，该数据应当小于256个字节
返回值	无
使用案例	PG2K_SSPI_WriteAPage(unsigned int PageAddr,unsigned char *datain)

用于编程嵌入式Flash普通页，时序图如下图所示。

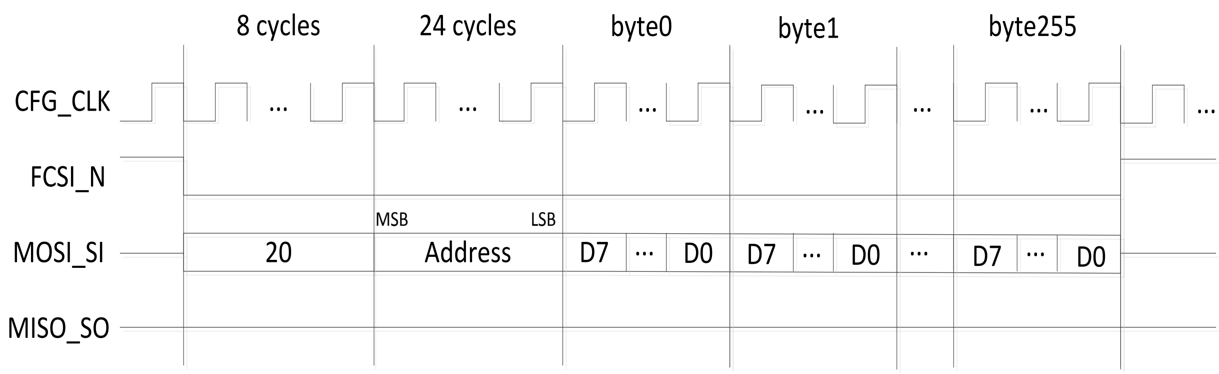


图 18 PROGRAM时序图

3.5.2 应用函数

应用类函数为底层功能函数的直接应用，这些函数与《Compact 系列 CPLD配置（configuration）用户指南》中附录4功能相对应。

3.5.2.1唤醒嵌入式FLASH

表 16 唤醒嵌入式FLASH

函数名	IntPG_SSPI_WakeUpEFLASH(void);
功能描述	唤醒当前嵌入式FLASH
输入参数说明	无
返回值	无
使用案例	参考demo程序的main.c
调用说明	使用RDID函数识别到需要配置的芯片后，进行休眠操作

唤醒嵌入式FLASH是执行写入UID，读写特征控制位、读写FLASH、读写主自加载位流等操作之前必须执行的一个操作，因此在执行上述操作之前，应当先检查状态寄存器，确认嵌入式FLASH是否处于休眠状态，如果处于休眠状态，则应当先唤醒，后执行。

CPLD在每次上电并完成主自加载位流之后，将会主动休眠嵌入式FLASH，优化功耗，在其他对降低功耗有要求的场合，如果不使用嵌入式FLASH，也建议将其休眠。

3.5.2.2 休眠嵌入式FLASH

表 17 休眠嵌入式FLASH

函数名	intPG_SSPI_SleepEFLASH(void);
功能描述	函数用于休眠嵌入式FLASH
输入参数说明	
返回值	无
使用案例	参考demo程序的main.c
调用说明	使用RDID函数识别到需要配置的芯片后，进行休眠操作

3.5.2.3读UID

表 18 读UID

函数名	unsigned intPG_SSPI_ReadOutUIDHigh(void); unsigned int PG_SSPI_ReadOutUIDLow(void);
功能描述	读取UID浩
输入参数说明	无

返回值	当前芯片的高32位和低32位 注：8bit为高位，所以，PG_SSPI_ReadOutUIDHigh(void)返还UID的高8位
使用案例	参考demo程序的main.c

3.5.2.4编程主自加载数据流

表 19 编程主自加载位流

函数名	Int PG_SSPI_Configuration(const char* cfg_file_name)
功能描述	写入主自加载位流
输入参数说明	此处参数位文件名，用于兼容有文件系统的嵌入式系统，例程中的调用不适用文件系统,置空即可，按如下方法调用： PG_SSPI_Configuration(“”);
返回值	写入成功 1 写入失败 0
使用案例	参考demo程序的main.c

3.5.2.5擦除主自加载数据流

表 20 擦除主自加载位流

函数名	PG2K_SSPI_EraseSomePage(unsigned int StartPage,unsigned int StopPage)
功能描述	擦除主自加载位流
输入参数说明	StartPage和StopPage分别为擦除的起始页、结束页。
返回值	无
使用案例	参考demo程序的main.c

擦除主自加载位流本质上也是按页擦除嵌入式FLASH操作的一种特殊形式，使用本函数时，将位流的起始和结束页面输入本函数，就会擦除指定页面的数据。

4 附录

表 21 Compact系列CPLD的SPI接口编号

Device	Package	Name	Pin Number
PGC1KG	MBG256	DIFFI_B2_1N/MOSI_SI	P13
		DIFFI_B2_21N/MISO_SO	T6
		DIFFI_B2_1P/FCSI_N	R12
		DIFFI_B2_21P/CFG_CLK	P6
	FBG256	DIFFI_B2_1N/MOSI_SI	P13
		DIFFI_B2_21N/MISO_SO	T6
		DIFFI_B2_1P/FCSI_N	R12
		DIFFI_B2_21P/CFG_CLK	P6
	LPG144	DIFFI_B2_1N/MOSI_SI	71
		DIFFI_B2_21N/MISO_SO	45
		DIFFI_B2_1P/FCSI_N	70
		DIFFI_B2_21P/CFG_CLK	44
	LPG100	DIFFI_B2_1N/MOSI_SI	49
		DIFFI_B2_21N/MISO_SO	32
		DIFFI_B2_1P/FCSI_N	48
		DIFFI_B2_21P/CFG_CLK	31
PGC1KL	UWG36	DIFFI_B2_0N/MOSI_SI	F1
		DIFFI_B2_10N/MISO_SO	E4
		DIFFI_B2_0P/FCSI_N	E1
		DIFFI_B2_10P/CFG_CLK	F5
PGC2KG	MBG256	DIFFI_B2_1N/MOSI_SI	P13
		DIFFI_B2_21N/MISO_SO	T6
		DIFFI_B2_1P/FCSI_N	R12
		DIFFI_B2_21P/CFG_CLK	P6
	FBG256	DIFFI_B2_1N/MOSI_SI	P13
		DIFFI_B2_21N/MISO_SO	T6
		DIFFI_B2_1P/FCSI_N	R12
		DIFFI_B2_21P/CFG_CLK	P6
	LPG144	DIFFI_B2_1N/MOSI_SI	71
		DIFFI_B2_21N/MISO_SO	45
		DIFFI_B2_1P/FCSI_N	70
		DIFFI_B2_21P/CFG_CLK	44
	LPG100	DIFFI_B2_1N/MOSI_SI	49
		DIFFI_B2_21N/MISO_SO	32
		DIFFI_B2_1P/FCSI_N	48
		DIFFI_B2_21P/CFG_CLK	31
PGC2KL	UWG49	DIFFI_B2_1N/MOSI_SI	G1
		DIFFI_B2_21N/MISO_SO	F5
		DIFFI_B2_1P/FCSI_N	G2
		DIFFI_B2_21P/CFG_CLK	F6
	SSBG256	DIFFI_B2_1N/MOSI_SI	N5
		DIFFI_B2_21N/MISO_SO	R11
		DIFFI_B2_1P/FCSI_N	M5
		DIFFI_B2_21P/CFG_CLK	T12

Device	Package	Name	Pin Number
PGC7KD	FBG484	DIFFI_B2_1N/MOSI_SI	W19
		DIFFI_B2_31N/MISO_SO	Y7
		DIFFI_B2_1P/FCSI_N	AA20
		DIFFI_B2_31P/CFG_CLK	AB6
	MBG400	DIFFI_B2_1N/MOSI_SI	W20
		DIFFI_B2_31N/MISO_SO	Y6
		DIFFI_B2_1P/FCSI_N	Y20
		DIFFI_B2_31P/CFG_CLK	W6
	MBG256	DIFFI_B2_1N/MOSI_SI	P13
		DIFFI_B2_31N/MISO_SO	T6
		DIFFI_B2_1P/FCSI_N	R12
		DIFFI_B2_31P/CFG_CLK	P6
	MBG332	DIFFI_B2_1N/MOSI_SI	V17
		DIFFI_B2_31N/MISO_SO	W7
		DIFFI_B2_1P/FCSI_N	W17
		DIFFI_B2_31P/CFG_CLK	V7
	LPG144	DIFFI_B2_1N/MOSI_SI	71
		DIFFI_B2_31N/MISO_SO	45
		DIFFI_B2_1P/FCSI_N	70
		DIFFI_B2_31P/CFG_CLK	44
PGC4KD	FBG256	DIFFI_B2_27N/MISO_SO	T6
		DIFFI_B2_27P/CFG_CLK	P6
		DIFFI_B2_1N/MOSI_SI	P13
		DIFFI_B2_1P/FCSI_N	R12
	MBG256	DIFFI_B2_27N/MISO_SO	T6
		DIFFI_B2_27P/CFG_CLK	P6
		DIFFI_B2_1N/MOSI_SI	P13
		DIFFI_B2_1P/FCSI_N	R12
	MBG324	DIFFI_B2_27N/MISO_SO	U6
		DIFFI_B2_27P/CFG_CLK	T6
		DIFFI_B2_1N/MOSI_SI	V17
		DIFFI_B2_1P/FCSI_N	U16
	MBG332	DIFFI_B2_27N/MISO_SO	W7
		DIFFI_B2_27P/CFG_CLK	V7
		DIFFI_B2_1N/MOSI_SI	V17
		DIFFI_B2_1P/FCSI_N	W17
	LPG144	DIFFI_B2_1N/MOSI_SI	71
		DIFFI_B2_31N/MISO_SO	45
		DIFFI_B2_1P/FCSI_N	70
		DIFFI_B2_31P/CFG_CLK	44
PGC4KL	UWG81	DIFFI_B2_1N/MOSI_SI	H1
		DIFFI_B2_27N/MISO_SO	J7
		DIFFI_B2_1P/FCSI_N	G1
		DIFFI_B2_27P/CFG_CLK	H7
	SSBG256	DIFFI_B2_1N/MOSI_SI	N5
		DIFFI_B2_21N/MISO_SO	R11
		DIFFI_B2_1P/FCSI_N	M5
		DIFFI_B2_21P/CFG_CLK	T12

Device	Package	Name	Pin Number
PGC4KLS	SBG81	DIFFI_B2_1N/MOSI_SI	H9
		DIFFI_B2_27N/MISO_SO	J4
		DIFFI_B2_1P/FCSI_N	J8
		DIFFI_B2_27P/CFG_CLK	J3
PGC10KD	MBG484	DIFFI_B2_1N/MOSI_SI	AA21
		DIFFI_B2_35N/MISO_SO	U9
		DIFFI_B2_1P/FCSI_N	AB21
		DIFFI_B2_35P/CFG_CLK	T9
	MBG400	DIFFI_B2_1N/MOSI_SI	W20
		DIFFI_B2_35N/MISO_SO	Y6
		DIFFI_B2_1P/FCSI_N	Y20
		DIFFI_B2_35P/CFG_CLK	W6

免责声明

免责声明

1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。

2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。

4、本文档披露的信息用于帮助用户解决应用方面的问题，但不保证文档没有瑕疵；如果用户没有按照本文档提供的方法操作，由此引起的功能异常和性能衰退，公司不承担任何责任，也无法承认是产品的问题。同时，本文档提供的方案仅是其中一种可行性方案，不能保证涵盖所有应用场景，如果用户按照本文档的描述操作存在功能异常或者性能衰退，公司也不能保证是产品的问题。